# Lab: Strings, Text Processing and Regular Expressions

Test your tasks in the Judge system:   
<https://alpha.judge.softuni.org/contests/strings-text-processing-and-regular-expressions-lab/4462>

## Reverse Strings

Write a program that:

* Reads a **series of strings** from the console, until you receive an **"end"** command
* **Reverse** given strings
* Print each **pair (old text and reversed text)** on a separate line in the format:

**"{word} = {reversed word}"**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| helLo  Softuni  bottle  end | helLo = oLleh  Softuni = inutfoS  bottle = elttob |
| Dog  caT  chAir  end | Dog = goD  caT = Tac  chAir = riAhc |

## Repeat Strings

Write a program that:

* Reads an **array of strings** from the console
* Each string is repeated **N times**, where **N is the length of the string**
* Print the **concatenated string**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| hi abc add | hihiabcabcabcaddaddadd |
| work | workworkworkwork |
| ball | ballballballball |

## Substring

Write a program that:

* Read **first string** from the first line of the console
* Reads **second string** from the second line of the console
* **Remove** **all** of the **occurrences** of the **first** **string** **in** **the** **second string**
* **Print** the **remaining string**

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| ice  kicegiciceeb | kgb | We remove "**ice**" once and we get "**kgiciceeb**"  We match "**ice**" one more time and we get "kgiceb"  There is one more match. The finam result is "kgb" |
| hep  ShepoftunihepIsGrhepeat | SoftuniIsGreat |  |

## Text Filter

Create a program that takes a **text** and a **string of banned words**. All words included in the ban list should be replaced with a string of **asterisks** '**\***', whose length must be equal to the word's length. The entries in the ban list will be separated by a **comma** and **space** "**,** ". The ban list should be entered on the first input line and the text on the second input line.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Linux, Windows  It is not **Linux**, it is GNU/**Linux**. **Linux** is merely the kernel, while GNU adds the functionality. Therefore we owe it to them by calling the OS GNU/**Linux**! Sincerely, a **Windows** client | It is not \*\*\*\*\*, it is GNU/\*\*\*\*\*. \*\*\*\*\* is merely the kernel, while GNU adds the functionality. Therefore we owe it to them by calling the OS GNU/\*\*\*\*\*! Sincerely, a \*\*\*\*\*\*\* client |
| von Richthofen, German, 80 air  Manfred Albrecht Freiherr **von Richthofen**, known in English as Baron **von Richthofen** was a fighter pilot with the **German** Air Force during World War I. He is considered the ace-of-aces of the war, being officially credited with **80 air** combat victories. | Manfred Albrecht Freiherr \*\*\*\*\*\*\*\*\*\*\*\*\*\*, known in English as Baron \*\*\*\*\*\*\*\*\*\*\*\*\*\* was a fighter pilot with the \*\*\*\*\*\* Air Force during World War I. He is considered the ace-of-aces of the war, being officially credited with \*\*\*\*\*\* combat victories. |

## Digits, Letters and Others

Write a program that:

* Read a **single** **string** from the console
* Print **all the digits** on the first line
* Print **all the letters** on the second line
* Print **all the other characters** on the third line

**Note:** There will always be at least one digit, one letter and one other character.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Agd#53Dfg^&4F53 | 53453  AgdDfgF  #^& |
| So%f94t34U\*n&i></37 | 943437  SoftUni  %\*&></ |

## Match Full Name

Write a program that:

* Reads a **text** from the console
* **Match full names** from the **given text**
* A **valid full name** has the following characteristics:
  + It consists of **two words**
  + Each word **starts** with a **capital letter**
  + After the first letter, it **only contains lowercase letters afterward**
  + **Each** of the **two words** should be **at least two letters long**
  + The **two words** are **separated** by a **single space**
* **Print full names** on the console, separated by single space

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Bethany Taylor, Oliver miller, sophia Johnson, SARah Wilson, John Smith, Sam Smith | Bethany Taylor John Smith |
| Elvis Presley a a C C-Muhammad Ali EE PeterpeterJR-a a xi ban D D bb b b-B B-c c EE-Michael Jackson DD peter smith B B PETER BROWN IVAN DAVIES- | Elvis Presley Muhammad Ali Michael Jackson |

## Match Phone Number

Write a program that:

* Reads a **text** from the console
* Create a **regular expression** to match a **valid phone number** from **Sofia**
* A **valid number** has the following characteristics:
* It starts with "**+359**"
* Then, it is followed by the area code (always **2**)
* After that, it's followed by the **number** itself:
  + The number consists of **7 digits** (separated into **two** **groups** of **3** and **4** **digits** respectively)
* The different **parts** are **separated** by **either a space or a hyphen** ('**-**')
* **Print all valid numbers** on the console, separated by a **comma and a space** ", "

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| +359 2 222 2222,359-2-222-2222, +359/2/222/2222, +359-2 222 2222 +359 2-222-2222, +359-2-222-222, +359-2-222-22222 +359-2-222-2222 | +359 2 222 2222, +359-2-222-2222 |
| +359 2 234 2324, 359-2-111-9876, +359/8/655/5432, +359-2 222 2222, +359 2-222-2222, +359-2-234-345, +359-2-123-45678, +359-2-222-2222, +359 2 654 1234 | +359 2 234 2324, +359-2-222-2222, +359 2 654 1234 |

## Match Dates

Write a program that:

* Reads a **text** from the console
* Create a **regular expression** to match a **valid dates**
* Every **valid date** has the following characteristics:
  + Format: "dd{separator}MMM{separator}yyyy"
  + Always starts with **two digits**, followed by a **separator**
  + After that, it has **one uppercase** and **two lowercase** letters (e.g. Jan, Mar)
  + After that, it has a **separator** and **exactly 4 digits** (for the year)
  + The separator could be either of three things: a period (**'**.**'**), a hyphen (**'**-**'**) or a forward-slash (**'**/**'**)
  + The separator needs to be **the same** for the whole date (e.g. 13.03.2016 is valid, 13.03/2016 is **NOT**). Use a **group backreference** to check for this
* Use **named** **capturing groups** in your regular expression.
* **Print all valid dates** on the console, separated by a **new line**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 13/Jul/1928, 10-Nov-1934, , 01/Jan-1951,f 25.Dec.1937 23/09/1973, 1/Feb/2016 | Day: 13, Month: Jul, Year: 1928  Day: 10, Month: Nov, Year: 1934  Day: 25, Month: Dec, Year: 1937 |
| 03-Mar-1878, 25/Apr/1915, 31-May-1916, 22/Jun-1941, 25.Dec.1937, 23/09/1973 | Day: 03, Month: Mar, Year: 1878  Day: 25, Month: Apr, Year: 1915  Day: 31, Month: May, Year: 1916  Day: 25, Month: Dec, Year: 1937 |